# Physical Design for Document Stores

**Moditha Hewasinghage, Alberto Abelló, Jovan Varga**
Universitat Politècnica de Catalunya, BarcelonaTech
[moditha | aabello | jvarga ]@essi.upc.edu

**Esteban Zimányi**
Université Libre de Bruxelles, Bruxelles, Belgium
ezimanyi@ulb.ac.be

## *Difficulties of Data Design*

*Physical-level queries*

- Multiple ways to design the data
  - Which kind of data store ?
  - Which logical design ?
- The design affects performance
- Decision is not trivial

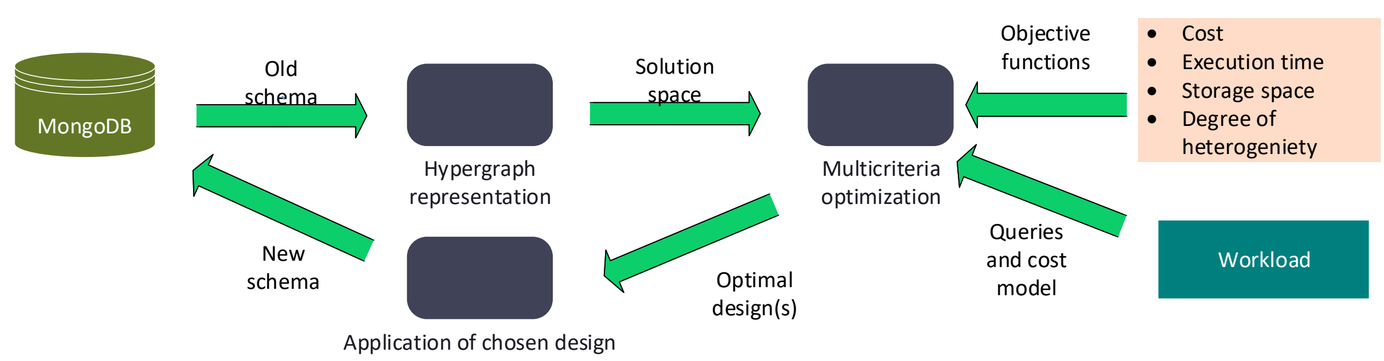
## *Goal*

Finding the optimal data design (s) for a given dataset and query load

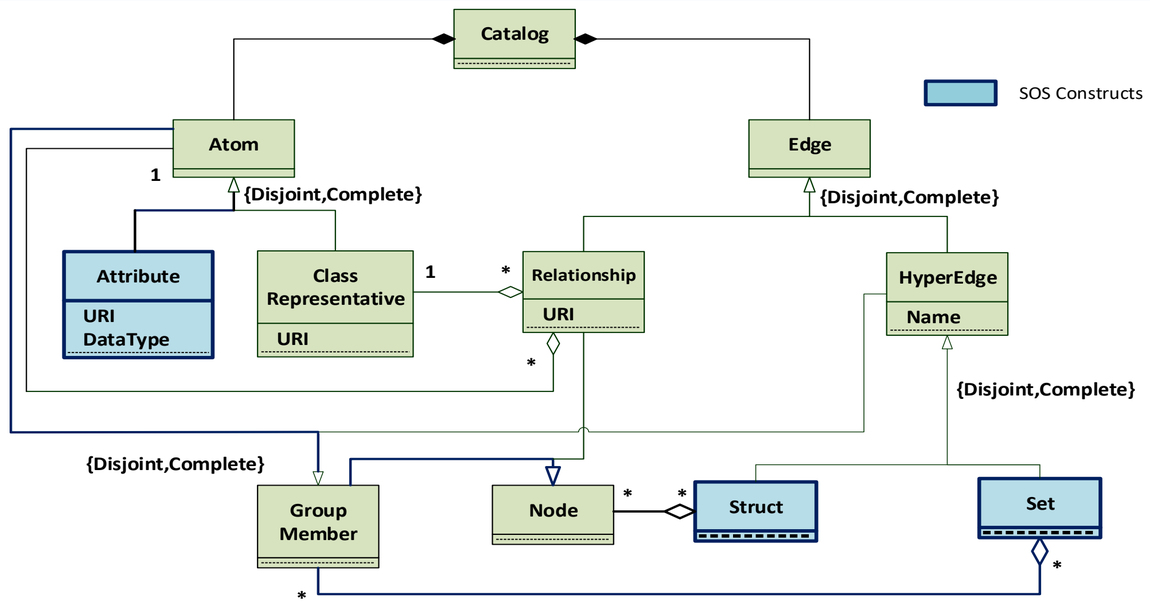
## *Problem as Multicriteria Optimization*

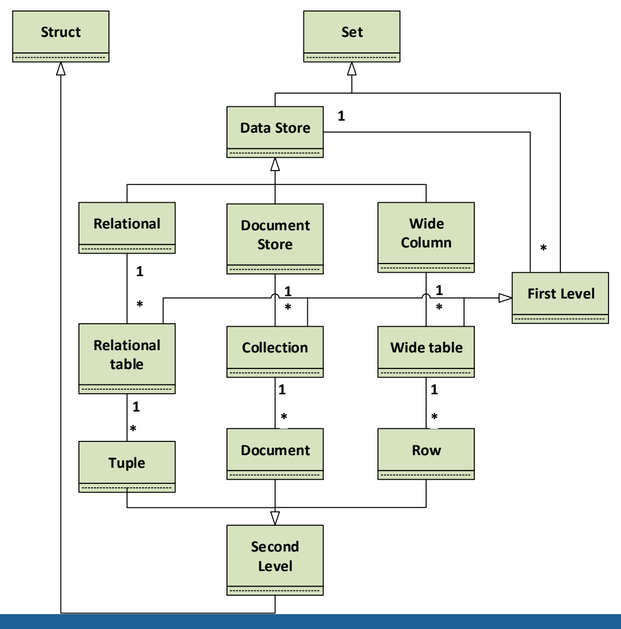
- Large solution space
  - Alternative data designs consisting of
    - Data store model
    - Logical design
- Contradicting Objective functions to minimize
  - Storage space
  - Query cost (CPU, disk I/O, memory)
  - Query execution time
  - Degree of heterogeneity

## *WorkFlow of the Approach (Focus on Document Stores)*



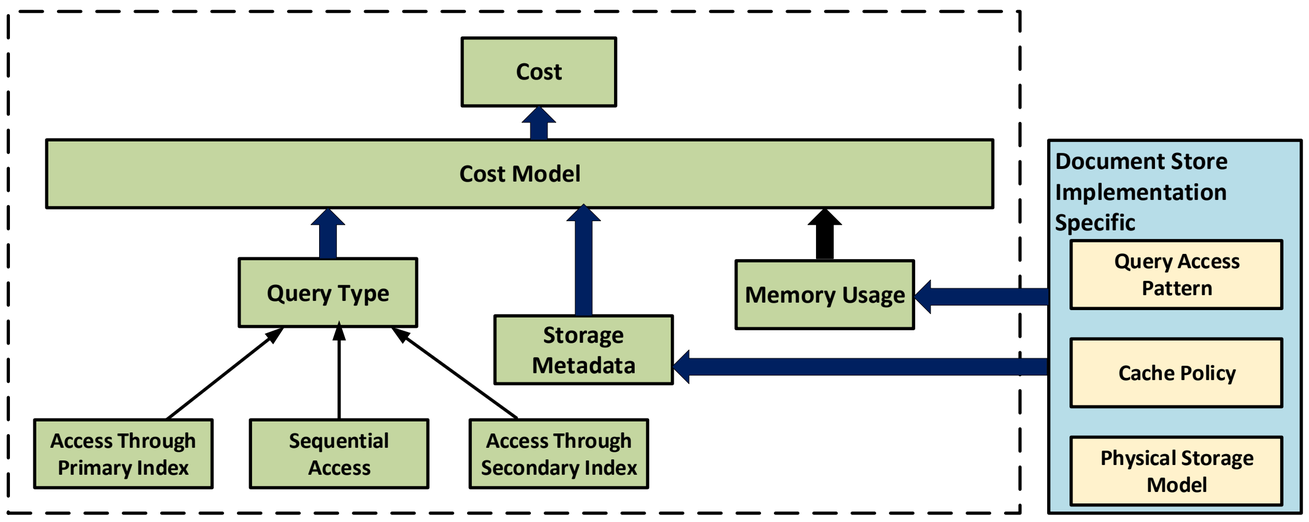

## *Hypergraph Representation as a Canonical Design Model [1]*




- RDF exemplars in a graph
- Build generalized hypergraph representing different design constructs
- Represent heterogeneous data models
  - Relational
  - Document Store
  - Column Family
- Identified constraints over different data models
- Simple query generation over the design
- Schema operations in the solution space for alternative designs (transformations)
- Modify the query algorithm to calculate other measures (size, frequency, runtime)

## *Cost model for Document Stores*




- No existing cost mode for document stores
  - Primitive approaches for query processing
- Disk I/O bases
- Random access queries
- MongoDB and Couchbase
  - Predefined memory size in Couchbase (buckets) LRU-like cache policy in MongoDB
    - Cache policy biased towards collection name (https://jira.mongodb.org/browse/WT-4732 Fix released in WiredTiger 3.2.1)
- Affected by
  - Document size
  - Document count
  - Access frequency
- Extend the cost model for secondary index

| | |
|---|---|
| $M$ | Total available memory |
| $Bsize_d$ | Block size for data |
| $Bsize_i$ | Block size for index |
| $T_m$ | Time to read a block from cache |
| $T_d$ | Time to read a block from disk |
| $C$ | A collection |
| $Size_d(C)$ | Average document size of a collection |
| $Size_i(C)$ | Average index entry size of a collection |
| $\|C\|$ | Number of documents in a collections |
| $R_d(C)$ | Average number of documents in a block |
| $R_i(C)$ | Average number of index entries in a block |

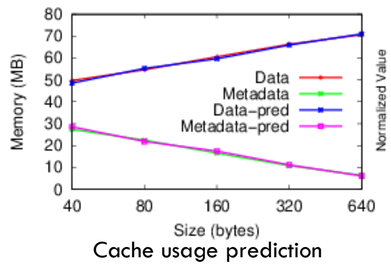
$$R_d(C) = f.\left\lfloor \frac{Bsize_d}{Size_d(C)} \right\rfloor \qquad R_i(C) = f.\left\lfloor \frac{Bsize_i}{Size_i(C)} \right\rfloor$$

$$B_d(C) = \left\lceil \frac{|C|}{R_d(C)} \right\rceil \qquad B_i(C) = \left\lceil \frac{|C|}{R_i(C)} \right\rceil$$

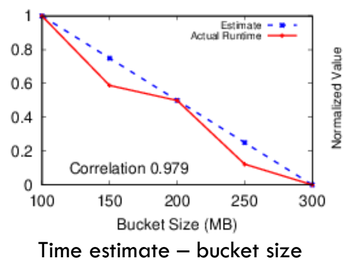
$$P_d(C) = \frac{M_d(C)}{B_d(C)} \qquad P_i(C) = \frac{M_i(C)}{B_i(C)}$$

$$Cost_{Rand} = \frac{T_m.(P_i(C) + P_d(C)) + T_d(2 - (P_i(C) + P_d(C)))}{2}$$

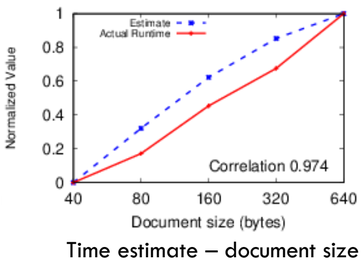
### Couchbase




Cache usage prediction

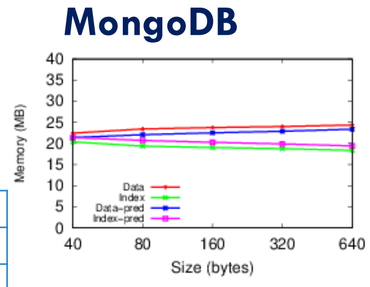


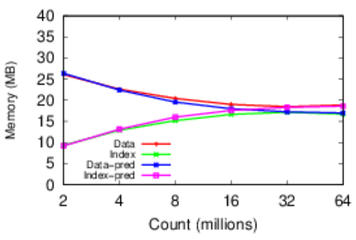

Time estimate – bucket size

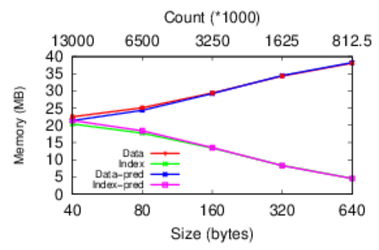



Time estimate – document size

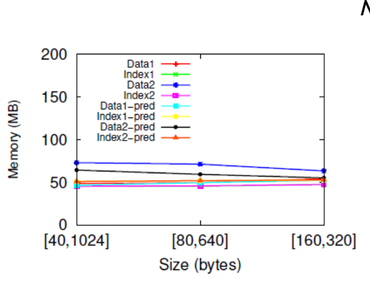
### MongoDB

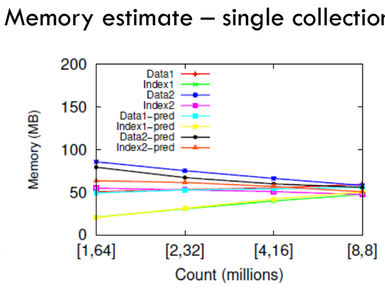


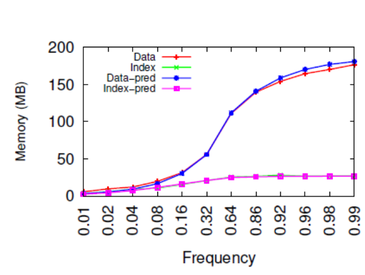



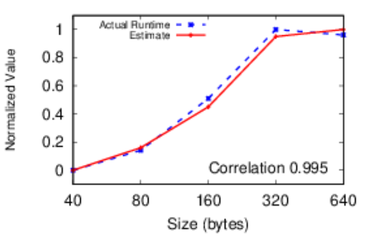

Memory estimate – single collection

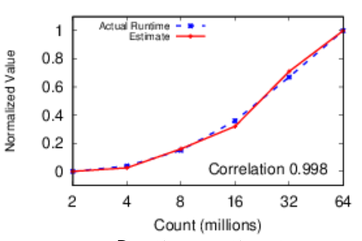


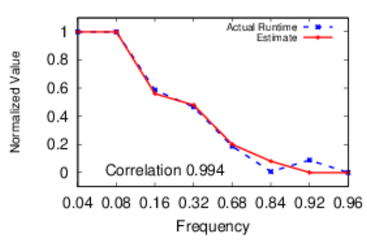




Memory estimate – two collections




Runtime estimate

## *References*

1. M. Hewasinghage, J. Varga, A. Abelló, and E. Zimányi. Managing Polyglot Systems Metadata with Hypergraphs. In International Conference on Conceptual Modeling. ER, 2018.
2. M. Hewasinghage, A. Abelló, J. Varga, and E. Zimányi. A Cost Model for Queries in Document Stores. In International Conference on Data Engineering. ICDE, 2020 (Under review)